UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/751,762	12/29/2000	Sailesh Kottapalli	2207/10121	5066

7590 06/03/2004

Kenyon & Kenyon
Suite 600
333 W. San Carlos Street
San Jose, CA 95110-2711

EXAMINER	
GERSTL, SHANE F	
ART UNIT	**PAPER NUMBER**
2183	8

DATE MAILED: 06/03/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/751,762	KOTTAPALLI ET AL.
	Examiner	Art Unit
	Shane F Gerstl	2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 March 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *12 March 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☒ Interview Summary (PTO-413) Paper No(s)/Mail Date. 8____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-21 have been examined.

Papers Received

2. Receipt is acknowledged of amendment and formal drawings papers submitted, where the papers have been placed of record in the file.

3. The amendment and arguments filed 12 March 2004 have successfully overcome the objection to the drawings, the title, the objections to the claims, and the 35 USC 112 rejections.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Parady (5,933,627) in view of Hennessy (Computer Organization and Design).

6. In regard to claim 1,

a. Parady discloses a method of handling operations in a multi-threaded processing system (figures 1 and 3), comprising:

i. determining if a stalled operation of a first thread is due to a loading of data from a memory device; Column 4, lines 57-59, show that blocking loads exist, which stall the operation on a memory access. Column 4,

lines 59-62 then show that a thread switch occurs on this blocking load waiting for data.

b. Parady does not disclose

ii. flushing an instruction from said first thread from a pipeline of said processing system when data is to be loaded from said memory device before executing said instruction.

c. Hennessy has shown flushing an instruction from a pipeline of said processing system. Page 505 of Hennessy shows that flushing is done in various pipeline stages to get rid of invalid instructions on a transfer of control such as an exception or branch or transfer of control to another thread as is the case in Parady as shown above.

d. Page 505 of Hennessy shows that transferring control (such as from one thread to another in the case of Parady) immediately (by flushing) is desirable so that invalid values (from invalid instructions) do not contaminate registers or memory locations. This ability to not contaminate registers and memory would have motivated one of ordinary skill in the art to modify the design of Parady to flush instructions on a transfer of control as taught by Hennessy. With this modification in place, Parady would flush instructions stored in the dispatch unit (column 3, lines 12-14 show that this unit holds up to four decoded instructions for execution) so they are not advanced for execution and contaminate memory and registers, as described by Hennessy, when Parady switches (transfers) to another flow of control of thread. Parady has further shown in column 4, lines

45-51 an embodiment where control is immediately transferred back to the thread that executed a load when the data is ready before other threads can advance. This means that an instruction about to be dispatched would have been flushed on a switch to a critical thread because data is to be loaded before the flushed instruction can execute.

It would have been obvious to one of ordinary skill in the art to modify the design of Parady to use the flushing technique described by Hennessy so that instructions not in the flow of control after a thread switch do not contaminate register and memory values.

7. In regard to claim 2, Parady in view of Hennessy discloses the method of claim 1, as described above, wherein said memory device is system memory coupled to a memory bus. Column 3, lines 58-61 show that a case for thread switching that involves a memory access would be when an L2 cache (figure 2, elements 80 and 82) misses. It is inherent that upon an L2 cache miss, data must be loaded from main memory. Figure 2 shows a memory bus, 86, that is coupled to main memory (system memory) that is used for getting data on a cache miss.

8. In regard to claim 3, Parady in view of Hennessy discloses the method of claim 1, as described above, further comprising: marking said instruction as a miss. Column 3, lines 58-61 show that a case for thread switching that involves a memory access would be when an L2 cache (figure 2, elements 80 and 82) misses. Thus the instruction is marked as a miss so the data can be retrieved from main memory.

9. In regard to claim 4, Parady in view of Hennessy discloses the method of claim 3, as described above, further comprising: rescheduling said instruction to be executed in

said pipeline. Column 4, lines 42-48 show that the thread and the instruction causing the switch is rescheduled for execution.

10. In regard to claim 5,

a. Parady discloses a method of handling operations in a multi-threaded processing system (figures 1 and 3), comprising:

i. determining if a stalled operation of a first thread is due to a loading of data from a memory device; Column 4, lines 57-59, show that blocking loads exist, which stall the operation on a memory access. Column 4, lines 59-62 then show that a thread switch occurs on this blocking load waiting for data.

b. Parady does not disclose

ii. flushing an instruction from said first thread from a pipeline of said processing system when data is to be loaded after a predetermined number of clock cycles from said memory device before executing said instruction.

c. Hennessy has shown flushing an instruction from a pipeline of said processing system. Page 505 of Hennessy shows that flushing is done in various pipeline stages to get rid of invalid instructions on a transfer of control such as an exception or branch or transfer of control to another thread as is the case in Parady as shown above.

d. Page 505 of Hennessy shows that transferring control (such as from one thread to another in the case of Parady) immediately (by flushing) is desirable so

that invalid values (from invalid instructions) do not contaminate registers or memory locations. This ability to not contaminate registers and memory would have motivated one of ordinary skill in the art to modify the design of Parady to flush instructions on a transfer of control as taught by Hennessy. With this modification in place, Parady would flush instructions stored in the dispatch unit (column 3, lines 12-14 show that this unit holds up to four decoded instructions for execution) so they are not advanced for execution and contaminate memory and registers, as described by Hennessy, when Parady switches (transfers) to another flow of control of thread. Parady has further shown in column 4, lines 45-51 an embodiment where control is immediately transferred back to the thread that executed a load when the data is ready before other threads can advance. This means that an instruction about to be dispatched would have been flushed on a switch to a critical thread because data is to be loaded before the flushed instruction can execute. Since a predetermined number of clock cycles is not further defined and could be any number, one can apply zero to be the predetermined number of clock cycles for instance. Also, there is inherently a predetermined number of cycles for loading data from memory. Therefore, the flushing is performed when data is to be loaded after a predetermined number of clock cycles from memory.

It would have been obvious to one of ordinary skill in the art to modify the design of Parady to use the flushing technique described by Hennessy so that instructions not in the flow of control after a thread switch do not contaminate register and memory values.

11. .In regard to claim 6, Parady in view of Hennessy discloses the method of claim 5, as described above, wherein said memory device is system memory coupled to a memory bus. Column 3, lines 58-61 show that a case for thread switching that involves a memory access would be when an L2 cache (figure 2, elements 80 and 82) misses. It is inherent that upon an L2 cache miss, data must be loaded from main memory. Figure 2 shows a memory bus, 86, that is coupled to main memory (system memory) that is used for getting data on a cache miss.

2. In regard to claim 7, Parady in view of Hennessy discloses the method of claim 6, as described above, further comprising: marking said instruction as a miss. Column 3, lines 58-61 show that a case for thread switching that involves a memory access would be when an L2 cache (figure 2, elements 80 and 82) misses. Thus the instruction is marked as a miss so the data can be retrieved from main memory.

3. In regard to claim 8, Parady in view of Hennessy discloses the method of claim 7, as described above, further comprising: rescheduling said instruction to be executed in said pipeline. Column 4, lines 42-48 show that the thread and the instruction causing the switch are rescheduled for execution.

4. In regard to claim 9, Parady in view of Hennessy discloses the method of claim 8, as described above, further comprising: executing said instruction when data is loaded from said memory device. Column 4, lines 42-48 show that a thread (and its stalled instruction) can be restarted as soon as the memory access is complete.

5. In regard to claim 10,

a. Parady discloses a processing system comprising:

i. a scheduler to pass instructions from a first thread and a second thread to an execution pipeline (figure 3, element 28);

ii. and pipeline control logic (figure 3, element 112 and figure 1, element 22) coupled to said execution pipeline to determine if a stalled execution of a first thread is due to a loading of data from a memory device. Column 4, lines 57-59, show that blocking loads exist, which stall the operation on a memory access. Column 4, lines 59-62 then show that a thread switch occurs on this blocking load waiting for data. Column 3, lines 58-61 show that a case for thread switching that involves a memory access would be when an L2 cache (figure 2, elements 80 and 82) misses. It is inherent that upon an L2 cache miss data must be loaded from memory. The thread switch logic will receive a signal from the cache showing a cache miss, as shown in figure 3, to show that a memory access is required and the thread switch logic will then carry out the flush.

b. Parady does not explicitly disclose a method to flush an instruction from said first thread from said execution pipeline when data is to be loaded from said memory device before said instruction can be executed.

c. Hennessy has shown flushing an instruction from a pipeline of said processing system. Page 505 of Hennessy shows that flushing is done in various pipeline stages to get rid of invalid instructions on a transfer of control such as an exception or branch or transfer of control to another thread as is the case in Parady as shown above.

d. Page 505 of Hennessy shows that transferring control (such as from one thread to another in the case of Parady) immediately (by flushing) is desirable so that invalid values (from invalid instructions) do not contaminate registers or memory locations. This ability to not contaminate registers and memory would have motivated one of ordinary skill in the art to modify the design of Parady to flush instructions on a transfer of control as taught by Hennessy. With this modification in place, Parady would flush instructions stored in the dispatch unit (column 3, lines 12-14 show that this unit holds up to four decoded instructions for execution) so they are not advanced for execution and contaminate memory and registers, as described by Hennessy, when Parady switches (transfers) to another flow of control of thread. Parady has further shown in column 4, lines 45-51 an embodiment where control is immediately transferred back to the thread that executed a load when the data is ready before other threads can advance. This means that an instruction about to be dispatched would have been flushed on a switch to a critical thread because data is to be loaded before the flushed instruction can execute.

It would have been obvious to one of ordinary skill in the art to modify the design of Parady to use the flushing technique described by Hennessy so that instructions not in the flow of control after a thread switch do not contaminate register and memory values.

6. In regard to claim 11, Parady in view of Hennessy discloses the processing system of claim 10, as described above, wherein said pipeline control logic is to mark said instruction as a miss. As shown above, the pipeline control logic includes the

cache control. Thus on a cache miss the instruction will be marked as a miss by the pipeline control logic.

7. In regard to claim 12, Parady in view of Hennessy discloses the processing system of claim 10, as described above, further comprising: an exception and retirement logic (figure 3, element 112) coupled to said execution pipeline.

8. In regard to claim 13, Parady in view of Hennessy discloses the processing system of claim 12, as described above, wherein said instruction marked as a miss is to be detected by said exception and retirement logic. As shown above, the pipeline control logic includes the cache control. Thus on a cache miss the instruction will be marked as a miss by the pipeline control logic. Also as shown above, the thread switching logic, receives this cache miss signal, or detects it. Thus the exception and retirement logic detects the instruction marked as a miss.

9. In regard to claim 14, Parady in view of Hennessy discloses the processing system of claim 13, as described above, further comprising: a fetch unit (figure 1, element 16) to provide said instruction to said scheduler.

10. In regard to claim 15, Parady in view of Hennessy discloses the processing system of claim 14, as described above, wherein said pipeline control logic is to cause said instruction to be executed when data is loaded from said memory device. Column 4, lines 42-48 show that a thread (and its stalled instruction) can be restarted as soon as the memory access is complete. Column 3, lines 54-56 shows that the threads pick up where left off upon the switch. Thus the thread control logic and further the pipeline control logic must be in control of the event.

11. In regard to claim 16,

a. Parady discloses a computing system comprising:

i. a memory bus (figure 3, element 114) coupled to system memory;

ii. and a processing system (figures 3 and 1) coupled to said memory bus, said processing system including

(1) a scheduler (figure 3, element 28) to pass instructions from first thread and second threads to an execution pipeline;

(2) and pipeline control logic (figure 3, element 112 and figure 1, element 22) coupled to said execution pipeline to determine if a stalled execution of a first thread is due to a loading of data from system memory. Column 4, lines 57-59, show that blocking loads exist, which stall the operation on a memory access. Column 4, lines 59-62 then show that a thread switch occurs on this blocking load waiting for data. Column 3, lines 58-61 show that a case for thread switching that involves a memory access would be when an L2 cache (figure 2, elements 80 and 82) misses. It is inherent that upon an L2 cache miss data must be loaded from main memory. The thread switch logic will receive a signal from the cache showing a cache miss, as shown in figure 3, to show that a memory access is required and the thread switch logic will then carry out the flush.

b. Hennessy has shown flushing an instruction from a pipeline of said processing system. Page 505 of Hennessy shows that flushing is done in

various pipeline stages to get rid of invalid instructions on a transfer of control such as an exception or branch or transfer of control to another thread as is the case in Parady as shown above.

c. Page 505 of Hennessy shows that transferring control (such as from one thread to another in the case of Parady) immediately (by flushing) is desirable so that invalid values (from invalid instructions) do not contaminate registers or memory locations. This ability to not contaminate registers and memory would have motivated one of ordinary skill in the art to modify the design of Parady to flush instructions on a transfer of control as taught by Hennessy. With this modification in place, Parady would flush instructions stored in the dispatch unit (column 3, lines 12-14 show that this unit holds up to four decoded instructions for execution) so they are not advanced for execution and contaminate memory and registers, as described by Hennessy, when Parady switches (transfers) to another flow of control of thread. Parady has further shown in column 4, lines 45-51 an embodiment where control is immediately transferred back to the thread that executed a load when the data is ready before other threads can advance. This means that an instruction about to be dispatched would have been flushed on a switch to a critical thread because data is to be loaded before the flushed instruction can execute.

It would have been obvious to one of ordinary skill in the art to modify the design of Parady to use the flushing technique described by Hennessy so that instructions not in the flow of control after a thread switch do not contaminate register and memory values.

12. In regard to claim 17, Parady in view of Hennessy discloses the computing system of claim 16, as described above, wherein said pipeline control logic is to mark said instruction as a miss. As shown above, the pipeline control logic includes the cache control. Thus on a cache miss the instruction will be marked as a miss by the pipeline control logic.

13. In regard to claim 18, Parady in view of Hennessy discloses the computing system of claim 16 wherein said processing system further includes an exception and retirement logic (figure 3, element 112) coupled to said execution pipeline.

14. In regard to claim 19, Parady in view of Hennessy discloses the computing system of claim 18, as described above, wherein said instruction marked as a miss is to be detected by said exception and retirement logic. As shown above, the pipeline control logic includes the cache control. Thus on a cache miss the instruction will be marked as a miss by the pipeline control logic. Also as shown above, the thread switching logic, receives this cache miss signal, or detects it. Thus the exception and retirement logic detects the instruction marked as a miss.

15. In regard to claim 20, Parady in view of Hennessy discloses the computing system of claim 19, as described above, wherein said processing system further includes a fetch unit (figure 1, element 16) to provide said instruction to said scheduler. In regard to claim 21, Parady in view of Hennessy discloses the computing system of claim 20, as described above, wherein said pipeline control logic is to cause said instruction to be executed when data is loaded from said system memory. Column 4, lines 42-48 show that a thread (and its stalled instruction) can be restarted as soon as

the memory access is complete. Column 3, lines 54-56 shows that the threads pick up where left off upon the switch. Thus the thread control logic and further the pipeline control logic must be in control of the event.

Response to Arguments

16. Applicant's arguments, see pages 7-8, filed 12 March 2004, with respect to the rejection(s)of claim(s) 1-21 under 35 USC 102(b) have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made based on Parady in view of Hennessy for claims 1-21. Hennessy clearly shows that flushing instructions from a pipeline is well and known in the art and used so that invalid instructions aren't advanced, which would contaminate registers and memory.

17. The examiner would like to point out that the argument that Parady specifically states that an operation must wait to be pointed to again by a sort of scheduler before resuming execution of that thread is merely one embodiment of Parady's invention. As shown in the previous Action and above, there is another specific embodiment of the invention that utilizes critical threads where a thread is switched back to when data for a load is ready and does not have to wait for a scheduler to point back to the thread.

18. No arguments concerning the individual limitations of the dependent claims other than the limitations of the parent independent claims have been entered.

Conclusion

19. The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or

patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

20. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The prior art cited in the previous Office Action remains pertinent and is cited herein with this Action as well.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shane F Gerstl whose telephone number is (703)305-7305. The examiner can normally be reached on M-F 6:45-4:15 (First Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703)305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Shane F Gerstl
Examiner
Art Unit 2183

SFG
May 27, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100